CSC 108H1 F 2007 Test 1
Duration — 25 minutes
Aids allowed: none

Student Number: ⌊_,_,_,_,_,_,_,_,_⌋

Lab day, time, room: _____

Last Name: _____   **First Name:** _____

Lecture Section: L0101   **Instructor:** Campbell

---

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This midterm consists of 3 questions on 6 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* Comments are not required except where indicated, although they may help us mark your answers. They may also get you part marks if you can't figure out how to write the code.
If you use any space for rough work, indicate clearly what you want marked.

1: _____/ 5

2: _____/ 5

3: _____/ 8

TOTAL: _____/18

Question 1.   [5 marks]

Define a function called `cherries` that has one `float` parameter representing the weight of a cherry, and returns a `str` representing a size designation, according to the following formula:

Weight Range	Classification
$\geq$10.4	'9 Row'
8.6 - 10.4	'10 Row'
7.1 - 8.6	'11 Row'
<7.0	'12 Row'

If a cherry weight is exactly at a category boundary (8.6), it should be classified in the heavier designation. Assume the input is in the range 0.0 - 15.0. You do not need to write a `docstring` comment.

Question 2.   [5 MARKS]

Complete the following function according to its `docstring` description.

```
import picture

def make_button()
    '''Return a 100 pixel wide by 100 pixel high Picture of a red button
    with circular holes.  The picture consists of three circles:  the
    red button is at location (10, 10) with a diameter of 80.  The two
    holes are both black to match the background, have diameter 10, and
    are located at (60, 45) and (30, 45).'''
```

Question 3. [8 MARKS]

Write a program that prompts for a picture using `pick_a_file`, and then uses `raw_input` to prompt for three numbers. The first is the amount to subtract from every red value in the picture, the second is the amount to subtract from every green value in the picture, and the third is the amount to subtract from every blue value in the picture. Your program should subtract those amounts from every pixel in the picture. At the end of the program, show the resulting picture.

You may assume all three numbers will be integers in the range 0 to 255.

You must include a function named `reduce_colours`, which has four parameters, in this order:

- The `Picture` to modify.

- An `int` representing the red subtracter.

- An `int` representing the green subtracter.

- An `int` representing the blue subtracter.

Function `reduce_colours` does the work of subtracting from each of the pixel values in the picture. If a new colour value is less than 0, use 0 as the value. You do **not** need to write a docstring comment.

Short Python function/method descriptions:

```
__builtins__:
  max(a, b, c, ...) -> value
    With two or more arguments, return the largest argument.
  min(a, b, c, ...) -> value
    With two or more arguments, return the smallest argument.
  raw_input([prompt]) -> string
    Read a string from standard input.  The trailing newline is stripped.  The prompt string,
    if given, is printed without a trailing newline before reading.
float:
  float(x) -> floating point number
    Convert a string or number to a floating point number, if possible.
int:
  int(x) -> integer
    Convert a string or number to an integer, if possible.  A floating point argument
    will be truncated towards zero.
picture:
  add_oval_filled(picture, x, y, w, h, acolor) --> None
    Takes a picture, a starting (x, y) position (two numbers), and a width and height (two
    more numbers, four total) then draws a filled oval of the given width, height and color
    with the position (x, y) as the upper left corner.
  make_empty_picture(width, height) --> Picture
    Return a new blank picture width pixels across and height pixels down.
  pick_a_file() --> string
    Launch a file chooser and return a string containing the name of the file that was selected.
  make_picture(filename) --> Picture
    Create and return a picture from the contents of filename.
  get_height(picture) --> int
    Takes a picture as input and returns how many pixels high it is.
  get_width(picture) --> int
    Takes a picture as input and returns how many pixels wide it is.
  get_red(Pixel) --> int
    Return the value of red (between 0 and 255) in the given pixel.
  get_green(Pixel) --> int
    Return the value of green (between 0 and 255) in the given pixel.
  get_blue(Pixel) --> int
    Return the value of blue (between 0 and 255) in the given pixel.
  get_pixels(picture) --> list
    Takes a picture as input and returns the sequence of pixel objects in the picture.
  set_red(Pixel, int) --> None
    Set the red value of the pixel to the int value.
  set_green(Pixel, int) --> None
    Set the green value of the pixel to the int value.
  set_blue(Pixel, int) --> None
    Set the blue value of the pixel to the int value.
  show(picture)
    Displays the picture.
```

Last Name: _____ **First Name:** _____